

Design and implementation of Earliest point initial (EDF) algorithm Using ARM core processor

Shweta R.Gulkari¹, Prof. Dinesh Rotake²

¹(Department of Electronic engineering, G. H. Raisonni women college of Engineering, Nagpur, India)

²(Department of Electronic engineering, G.H.Raisonni women college of Engineering, Nagpur, India)

Abstract : This paper describe accept and the accomplishing of an Earliest point initial (EDF) founded algorithm to abutment amount of absolute time submissions on a abandoned processor. EDF is an optimal scheduling algorithm on preemptive uniprocessors. EDF can affirmation that all deadlines are met supplied that the absolute CPU appliance is not added than 100%. Contrasted to anchored antecedence alignment techniques like rate-monotonic of scheduling, EDF can assertive all the deadlines in the arrangement at college loading. EDF alignment in real-time accretion it is accessible to appraise inferior case acknowledgment times of methods in EDF, to accord with added kinds of processes than alternate methods and to use servers to adapt overloads. This action is to be accord the achievability to the next appointed for execution the after-effects apparent from simulation displays and ADC (Temperature sensor), RTC, UART (TX), UART (RX), LCD, RELAY. This following of task at any agreeable time instant is very effective $\mu\text{C}/\text{OS-II}$ activated in this algorithm for programs allotment device, is a multi-task real-time operation arrangement and along analyzes. The development adjustment which is real-time action arrangement $\mu\text{C}/\text{OS-II}$ directed in microcomputer accessory based on ARM. This (EDF) algorithm was implemented on development lath LPC2148 and outcomes were verified.

Keywords: EDF RTOS, Real-Time System, $\mu\text{C}/\text{OS-II}$

I. Introduction

Real-time systems may be outlined as those systems that respond to external events in an exceedingly timely fashion, the latent periodic bonded. This paper deals with the porting of $\mu\text{C}/\text{OS-II}$, a real time kernel in ARM battery-powered microcontroller for the implementation of multitasking and time planning. The heart of the system may be a time period kernel that uses preemptive planning to attain multitasking on hardware platform. The Arm processor gets all relevant data from one supply i.e. the link-attached terminal section. The temperature analysis information is premeditated in convenient digital display. Advanced real-time techniques software is applied, that are assiduously acclimated in analytical areas like space research and defense applications and also to realize an industrial real time parameter monitoring system. $\mu\text{C}/\text{OS-II}$ is the operating system of supporting many tasks, can support 64 tasks at most, different task carry out different functions; [1] $\mu\text{C}/\text{OSII}$ is a multi-task real-time action arrangement with accessible antecedent cipher, can coalesce can transplant, and can carve down. It is a preemptive real-time atom preemptive [2]. $\mu\text{C}/\text{OS-II}$ adopts semaphore, mailbox, calendar queue, accident banderole accumulation to advertisement and synchronizes. Its source code is well organized, understandable and can be adapted simply to the ARM core. In earliest point initial (EDF) arranging, it is an optimal activating capital affair Scheduling Algorithm. The abecedarian abstraction of this algorithm is simple to understand. The schedulability analysis for EDF is easy. A assignment is alone calendar beneath EDF, alone if it satisfies the cachet that absolute processor appliance (ui) due to the assignment set is beneath than 1. With alignment alternate processes that accept deadlines according to their periods, EDF has a appliance accountable of 100%. Thus, the schedulability analysis for EDF is: [3]

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1, \quad \longrightarrow \quad \boxed{1}$$

Where the $\{C_i\}$ are the worst-case computation-times of the n processes and the $\{T_i\}$ are their corresponding inter-arrival periods (assumed to be identical to the affiliation deadlines)

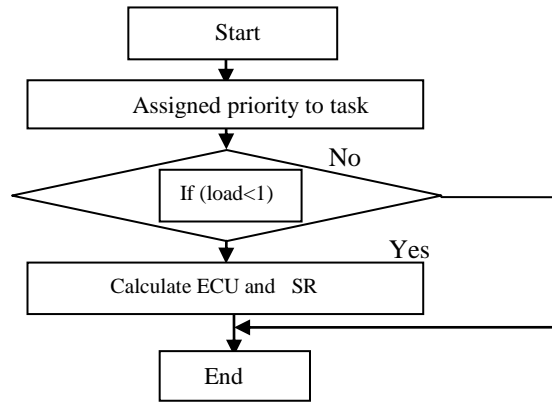


Fig.1 Basic flow diagram of EDF

This is the basic flow design drawing of EDF start the method and each task allotted main concern and then condition ascertain if burden less than 1 then it's calculate execution current unit or the condition if larger than 1 is does not assess the execution present unit is the exactly end . As deadline based algorithm is less predictable and less controllable. Most os available [4][5] in the market use priority based co-operative scheduling pre-emptive scheduling. $\mu\text{C}/\text{OS-II}$ [6] is a multitasking real-time kernel that provides an execution environment for many tasks, where every task can utilize system resources. It provides transfer of execution from one task to the other, so that resources can be used efficiently and timing deadlines can be achieved. $\mu\text{C}/\text{OS-II}$ supply down latencies for the kernel services. Each task is assigned a priority and is scheduled according to it. Furthermore, preemption is supported; a higher priority task may preempt execution of a lower priority task in order to perform a time-critical function. It is portable since it has been implemented mostly in ANSI C and contains only a small amount of assembly code for adaptation to a particular processor core. In fact, it has been ported to more than 40 different processor architectures ranging from 8- to 64- bit microprocessors, microcontrollers and digital signal processor [6]

II. Hardware Circuit

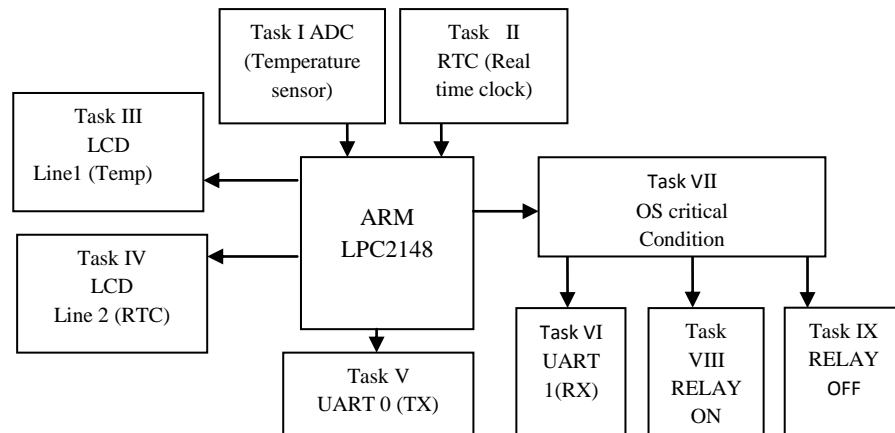


Fig 2.Overview of architecture:

Power supply of 3.3v is given to the ARM processor LPC2148.the surrounding temperature is using abstinent temperature sensor.LM35, the temperature sensor is connected to ADC0, input temperature value shows on LCD line 1 and same RTC value shows on LCD line 2 .UART0 (TX) is use for transmitting the value of Temperature and RTC and relay is use for note down temperature value n RTC value. os critical condition are set 50°C temperature if in case temperature are increase above 50°C that time this first five task are disable and other four task enable. Task eight (relay on) and task nine (relay off) is dependence on UART (RX).

III. Starting μ C/OS-II

In any application μ C/OS-II is started as shown in the figure 1. Initially the hardware and software are initialized. The hardware is the ARM core and software is the μ C/OS-II. The resources are allocated for the Tasks defined in the application. The scheduler is started then. It schedules the tasks in preemptive manner. All these are carried out using specified functions defined in μ C/OS-II.

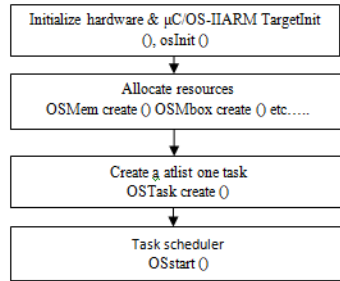


Figure 3: Starting μ C/OS-II

3.1 Initializing μ C/OS-II:

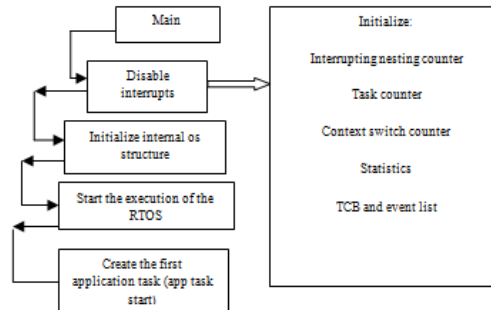


Figure 4: Initializing μ C/OS-II

Figure 4: Initializing μ C/OS-II

3.2 Task Creation:

To make it ready for perform multitasking, the kernel needs to have information about the task: its starting initial address, top-of-stack (TOS), priority, arguments passed to the task, related information about the task.

3.3 Implementation through μ C/OS-II:

In embedded systems, a board support package (BSP) is implementation specific support code for a given (device motherboard) board that conforms to a given operating system. It is commonly built with a boot loader that contains the minimal device support to load the operating system and device drivers for all the devices on the board.

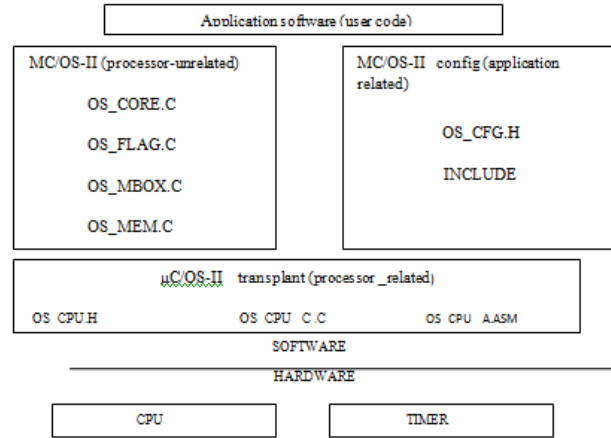


Fig.5 μC/OS-II hardware and software architecture

IV. Program Flowchart

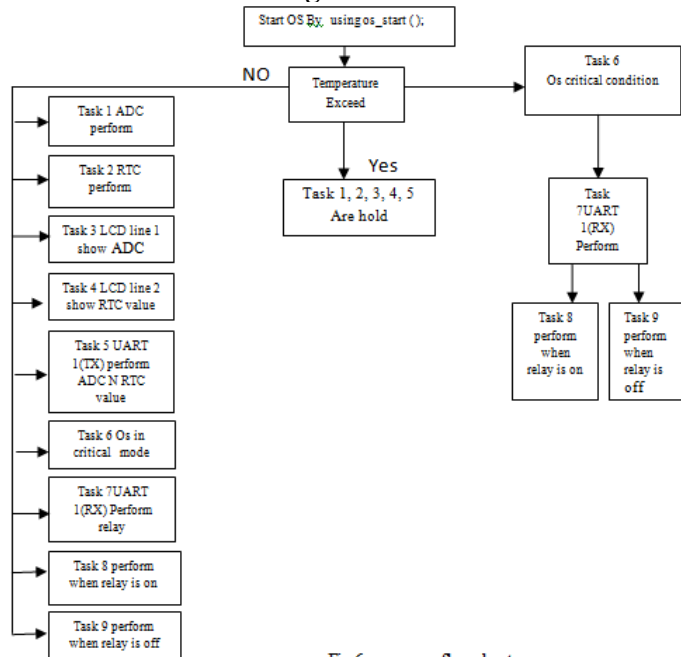
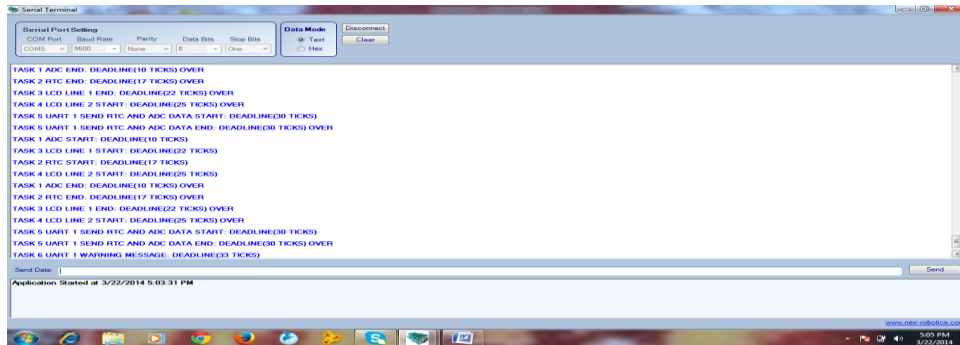


Fig.6 program flowchart

V. Result

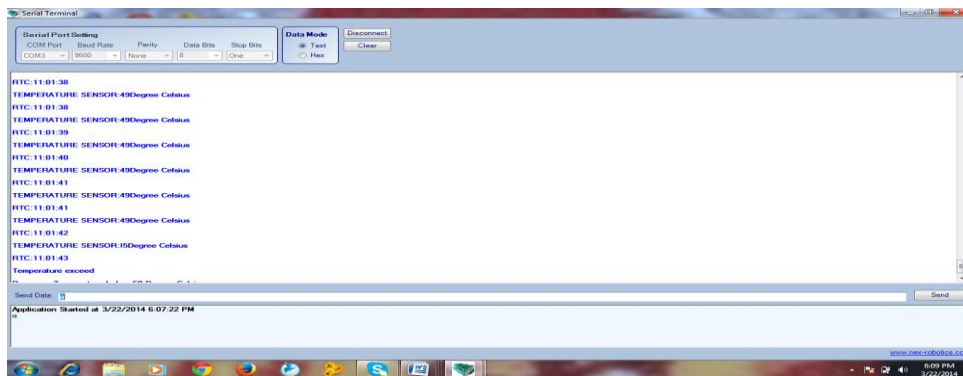
Hardware design is using ARM LPC 2148. Different tasks including temperature sensor, LCD, relay, RTC, are performed simultaneously through the μ C/OS-II programming with EDF algorithm. Program was done in keil 4 version. And task performs on serial terminal

Case1:



(a)

In case 1 (a) all task are perform simultaneously. And same time task 5 is sent the value of temperature sensor a RTC on second serial terminal are shown (b).



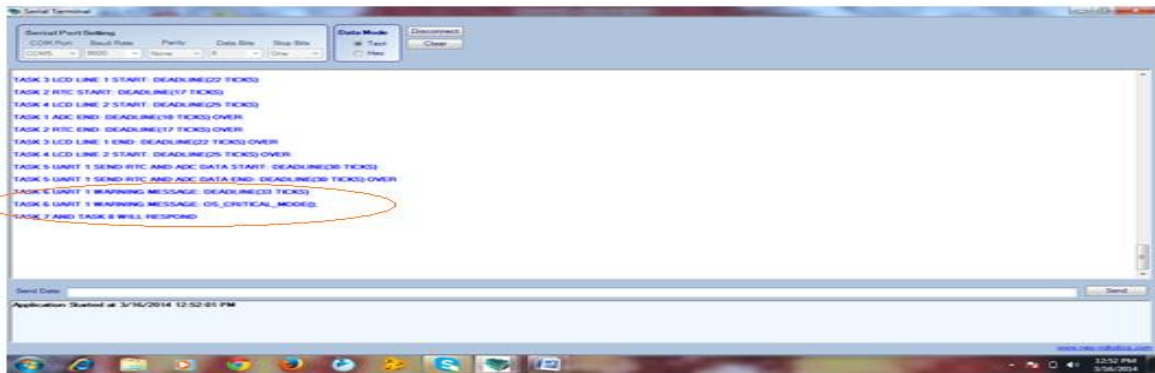
(b)

In this case1 (b) show temperature value and RTC value and (c) at this time same value show on LCD and perform intertask communication.



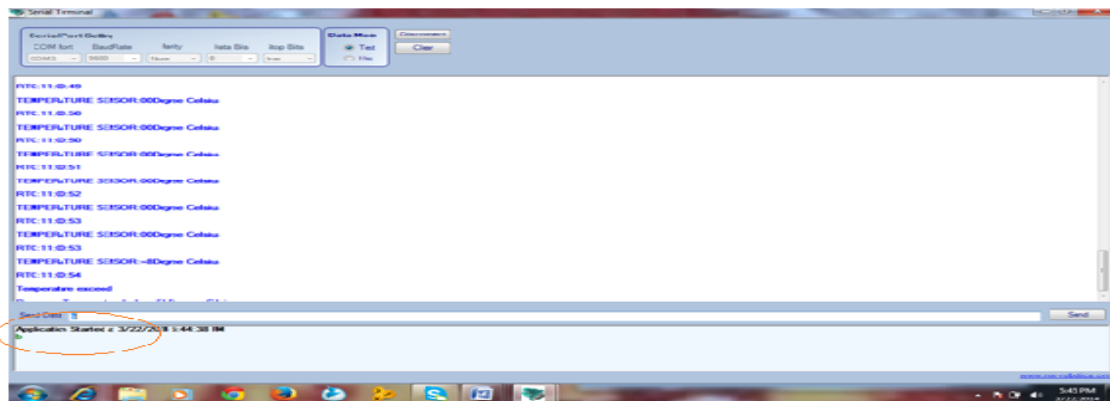
(c)

Case2:



(a)

In case 2 (a) when Temperature increases on 50°C at that time task 6 give the warning and os task in critical mode and this time all task are hold but 7,8,9 task are perform .

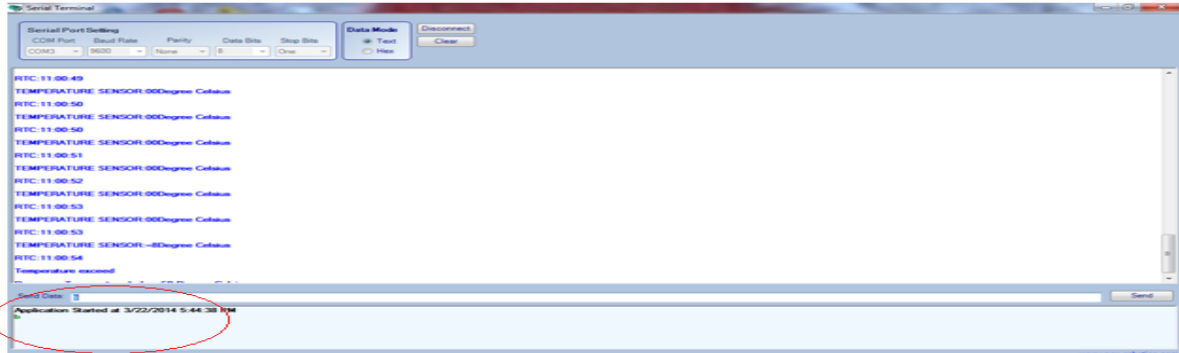


(b)

In this case 2 b fig show when we are writing 'a' on sent data that time task 7 is perform and (c) relay is 'ON' means task 8 are perform.



(c)



(d)

In that case 2 (d) when we are writing 'b' on the sent data that time task 7 is perform and (e) relay is 'OFF' means task 9 are perform



(e)

VI. Conclusion

EDF based scheduling has been proposed .simulation on carried out on $\mu\text{C}/\text{OS-II}$ to execute EDF algorithm for 9 tasks. Firstly, the need of a Real Time Operating System is analyzed and the advantages of $\mu\text{C}/\text{OS-II}$ RTOS for this paper was described. Secondly, porting of $\mu\text{C}/\text{OS-II}$ to ARM processor has been discussed in detail. Finally, using $\mu\text{C}/\text{OS-II}$ programming with EDF algorithm was successfully achieved for all 9 tasks. Results of practical application indicate that the proposed system has high performance, safety and reliability. In further study, we can cut, expand and optimize the system.

References

- [1]. Pinkesh Pachchigar , P Eswaran, Amol Kashinath Boke "Design and Implementation of Deadline based EDF Algorithm on ARM LPC2148," Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
- [2]. An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.2, February 2011
- [3]. James H. Anderson, Vasile Bud, and Uma Maheswari C. Devi "An EDF-based Scheduling Algorithm for Multiprocessor Soft Real-Time Systems" Proceedings of the 17th Euromicro Conference on Real Systems (ECRTS'05), 6-8 July, 2005, pp.199-208.
- [4]. Su-Lim TAN and Tran Nguyen Bao Anh , "Real-time operating system (RTOS) for small (16-bit) Microcontroller "proceedings The 13th IEEE International Symposium on Consumer Electronics (ISCE2009), pp. 1007-1011
- [5]. Liu Yong, Huang Xinbo "The Application of RTOS in Microcomputer Protection Device" International Conference on Electrical and Control Engineering 2010 pp 2264-2268.

Book :

- [6]. Jean j Labrosse, "Micro/ os-II the real time kernel".2nd Edition, cmp books, 2002.